

A Novel Approach for the Link Layer in Impulse-based UWB Ad Hoc Networks

NAN SHI, LIANG XIA and IGNAS G. NIEMEGERERS

Wireless and Mobile Communications (WMC) Group, Delft University of Technology, 2628CD, Delft, The Netherlands

E-mails: {n.shi, l.xia, i.niemegeers}@ewi.tudelft.nl

Abstract. Ultra Wide Band (UWB) impulse radio, promises to be suitable for short-range, low-power, low cost and high data rate applications. While most UWB research is concentrating on the physical layer, little research has been published on the link layer. The fundamental operations in the link layer need to take into account the specifics of impulse radio and also the particular features of ad hoc networks. A novel self-organizing link layer protocol based on time hopping impulse radio, called SDD, was proposed by the authors. This protocol is a collision-free mechanism that enables the devices to discover neighboring nodes and arrange the access to communication resources shared among the nodes. In this paper, some issues related to the self-organizing link layer based on UWB impulse radio are investigated and addressed. The SDD protocol is further developed and specified in detail. The simulations are carried out using GloMoSim simulation environment. An SDD module has been developed and embedded in the simulator. Results show that the SDD protocol can work properly and efficiently in a single-hop ad hoc network.

Keywords: impulse radio, time hopping, device discovery, ad hoc

1. Introduction

Future Personal Area Networks (PANs) should be able to support a large variety of personal applications. Some of the more demanding ones will be video conferencing, interactive games, home theatre, and high-resolution image transfer. Ultra Wide Band impulse radio (UWB-IR) promises to be suitable for such short-range, low-cost and high data rate applications while having a very low energy consumption. IEEE 802.15 is the standardization body which covers the link layer and physical layer technologies for PANs. UWB-IR is one of the technologies considered within IEEE 802.15.3a.

PANs that meet these expectations will have a hybrid character, consisting mainly of ad hoc networks, using IEEE 802.15 air-interfaces, with occasional access to infrastructures. Because of the mobility of a person and the devices that constitute a PAN, and the dynamics of the applications [1], nodes will join or leave the network, and radio links will be broken or established. Therefore, the network should be able to quickly configure and reconfigure itself. These operations should be done without the intervention of a user or a system administrator, and therefore will have to be self-organizing. Self-organization in this context implies the spontaneous discovery of neighboring nodes, the creation of connections, the scheduling of transmissions and the formation and re-configuration of the network topology [2].

In [3] we introduced a novel link layer protocol for PANs based on UWB-IR, i.e. Self-organizing Device discovery and Data transmission (SDD) protocol. The SDD protocol makes use of time hopping multiple access, which is an intrinsic feature of UWB-IR. This protocol is

able to automatically discover nodes within radio range, form a distributed link layer topology and assign channel resources for collision-free transmissions.

In this paper we highlight the key aspects in the design of the SDD protocol, present a further developed version of the protocol and analyze its behavior by means of simulation. The paper is organized as follows. In Section 2 we define several properties to characterize the quality of the self-organization process at the link layer. In Section 3, we discuss issues related to the design of SDD protocol, such as UWB-IR features, device discovery and code assignment. In Section 4, we describe the sub-processes and relative algorithms of the SDD protocol. In Section 5, the performance parameters which characterize the quality of self-organization in the SDD protocol are specified and analyzed by means of simulation. Conclusions are drawn in Section 6.

2. Quality of the Self-organization Process

The quality of the discovery processes in self-organization protocols in the link layer can be measured by the discovery time, the discovery ratio, and the effect of the discovery processes on network throughput. Moreover we will argue that robustness is an important characteristic for self-organization in a dynamic network environment. Let us discuss each of them in detail.

- *Discovery time.* An important measure is the average *discovery time* \bar{T}_{disc} . We define \bar{T}_{disc} as the average time spent by a self-organization protocol to enable any newly entering node or group of nodes to discover and incorporate with all or most neighboring nodes.

The *discovery time* should be fast enough to allow timely configuration and re-configuration of the link topology relevant to the dynamics of the network. Bluetooth random-access inquiry mechanisms are reputed to be slow with delays of the order of 10s [4]. The limited Bluetooth channel capacity is one reason for the long discovery time. The other reason is that the inquiry and inquiry scan procedures work only for two devices at a time so that a large number of inquires have to be performed by a node to discover all possible its neighbors.

A relevant measure is the time for a particular node i to detect a neighboring node j has joined or left, denoted by $T_{i,\text{join}}^j$ and $T_{i,\text{depart}}^j$.

- *Discovery ratio.* The capability of each node to gain information about its neighborhood is determined both by system requirements and discovery processes. Each node has its own view of its neighbors. A list with the neighbors' information is maintained in every node. We define the *discovery ratio* $\bar{\rho}$ as the average percent of the number of neighbors on the list in the total number of nodes within the radio range. Note that this measure relates to the network as a whole and not to an individual node. The discovery ratio determines the topology of a network. For the same number of in-range nodes, when the discovery ratio is higher, the nodes are more aware of the topology, i.e., the network is more strongly connected.

- *Effect on throughput.* When network nodes frequently join and leave, the self-organization processes, i.e., device discovery and connection setup, need to be performed more often. More control messages will consume the radio resources they share with the data transmission. The impact of the control message overhead on throughput should not be significant. We define the *effect on throughput* as the achievable throughput measured when discovery processes are performed to the throughput measured when discovery processes are not performed.

- *Robustness.* A robust topology should ensure that communication remains possible in spite of changes to the network [5]. In wireless ad hoc networks, the changes are link or node

failures caused by nodes' frequently joining and leaving. A node failure is defined as all links to the node failing at the same time.

For a network to be robust, it is beneficial that it has the property of node-similarity, which requires all nodes to be equally informed about the network topology [5]. A network without node similarity is not robust. For instance, the master node in a Bluetooth piconet controls the address and clock information of its slave nodes but not the other way around. When the master node leaves the network due to mobility or node failure, the slaves lose the communication capability with other slaves in the same piconet. A new piconet topology has to be constructed by starting a time-consuming discovery process from scratch.

In a node-similar network, each node has the same topology information as its neighbors. Routing possibilities can be fairly provided for each node based on the information and a more robust multi-hop network topology can be formed.

3. Design Considerations for UWB-IR Based Networks

In order to design a high data rate and fast medium access in UWB ad hoc network, a number of facts and design issues related to the UWB-IR technique need to be kept in mind.

3.1. UWB-IR CHARACTERISTICS

In current research, impulse radio and multi-band UWB systems are both being investigated. Time Hopping (TH) and Direction Sequence (DS) spread spectrum are considered in UWB-IR systems and Orthogonal Frequency Division Multiplexing (OFDM) is the candidate for multi-band UWB systems [6].

We consider TH impulse radio, where data can be modulated by pulse position. A low duty cycle and low power density are crucial properties of TH impulse radio. Multiple-access can be achieved by TH spread spectrum, which uses a pseudorandom time hopping sequence, called the TH code. TH impulse radio also has in principle the capability to determine its location with great accuracy, i.e., of the order of sub-centimeter [7]. Due to the low duty cycle of impulse radio, full-duplex can be achieved by blanking the receiver input during pulse transmission by any of the transmitter units in the same node [8].

3.2. MULTIPLE-ACCESS

When the TH multiple-access method is used, if simultaneous packets transmitted on different codes arrive, the node is able to decode the information on its own code and the information on other codes is perceived as noise [9]. Thus, the multiple-access interference (MAI) will be a limiting factor for the node capacity. There is a trade-off between the achievable data rate per node and the number of simultaneous nodes. However, in [10] it is shown that the multiple-access channel can achieve a full node capacity if the number of nodes is small. Under certain conditions (e.g., a high spreading ratio), it can be assumed that MAI has no effect on the per node capacity for less than 10 nodes [10]; the aggregate capacity can be simply calculated as the number of nodes times the full node capacity. We base our design on the assumption that at any time no more than 10 nodes communicate simultaneously.

Furthermore, when overlapping packets sent on the same code arrive (Figure 1), multiple pulses are received in one frame and the pulses may not exactly overlap. The result is an

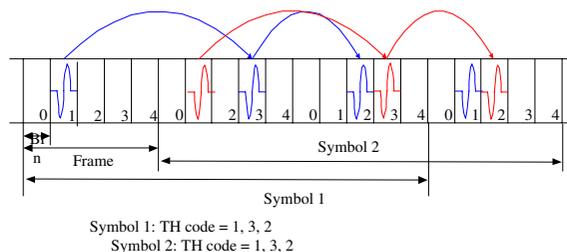


Figure 1. Packets overlapping on the same TH code.

increase in MAI but not a collision since packets are still successfully received. By using a novel blind synchronization method [11], the receiver is able to decode the overlapping packets of the same code. A packet collision happens only when two or more packets with the same code exactly coincide in time at a node; the physical layer will detect this collision.

Therefore, we can neglect the impact of MAI on our protocol design and ensure that the achievable data rate of a node is not affected by MAI.

The low power density and low duty cycle of TH impulse radio however implies that it is difficult to distinguish idle periods from packet transmission periods in an UWB receiver [12] and then a long channel detection delay cannot be avoided. The class of MAC protocols based on carrier-sensing, e.g., CSMA/CA, will result in poor performance in UWB ad hoc networks. Therefore, carrier-sensing MAC protocols should not be preferred.

The signal acquisition time of impulse radio receivers is longer than conventional receivers; can be of the order of milliseconds. The transmission of one data packet between two devices normally requires one or more signal acquisitions. The long signal acquisition time may degrade the achievable throughput [13]. To mitigate this effect we have a physical layer design which targets a relatively short acquisition time, of the order of 0.1 ms [11]. Moreover, the link layer protocol design should as much as possible avoid frequent synchronizations.

3.3. DEVICE DISCOVERY

Using TH spread spectrum as the multiple access method implies that data transmission is not possible before the communication codes are known by the sending and receiving nodes. Thus, device discovery, i.e., neighbor discovery, is a prerequisite. Neighbor discovery is the process during which nodes get to know the presence and identity (e.g., network ID, time/frequency hopping pattern) of their neighbors. Neighbor discovery is not only used to form a network in the initialization phase, but also to discover the existence of new neighbors and the loss of connectivity with existing neighbors.

3.4. CODES

TH impulse radio has a large and adjustable code family [10]. Unlike Bluetooth and 802.11 WLAN, where one code is shared with some or all of the adjacent nodes, TH impulse radio enables each individual node to be assigned different codes. A code assigned to each user contains a TH sequence [11]. The TH sequence is a distinctive time-shift pattern to eliminate collisions due to multiple access.

We define three basic types of codes: *common*, *transmitter-based* and *receiver-based* codes [5], denoted, respectively, by C , $C_{i,T}$ and $C_{i,R}$. The receiver-based and transmitter-based codes

can be derived from a node's unique 48-bit IEEE MAC address by using a pseudo random process, e.g., using the method proposed in [14].

The common code is defined as a signaling channel for exchanging specific control messages. The common code is a fixed short access code monitored by all idle nodes for packet arrivals. It is used to initialize a neighbor discovery or a data transmission through a broadcast operation. Once a communication is finished, nodes need to monitor the common code for the next packet arrival.

We explained in Section 3.2 that, when control messages on the common code partly overlap in time, the communications between different idle node pairs can still successfully take place, albeit at the expense of higher MAI. To analyze the collision probability on common code, an approximate analysis is used. We assume the data rate is 110Mbps and two control packets with a length of 36 bytes overlap. A collision happens when two packets only have less than one pulse time difference. The probability of a collision can be roughly calculated as $\frac{10^{-9} \times 110 \times 10^6}{36 \times 8} = 3.82 \times 10^{-4}$, which is negligible.

3.5. ROBUST LINK TOPOLOGY

Robustness is a requirement in a dynamic ad hoc network where nodes frequently become unreachable or stop functioning. In this paper, we consider single-hop networks where nodes are within radio range of each other. Each node fairly keeps the link topology information. Therefore a node does not depend on information stored in an adjacent node for knowing its connectivity. This makes the network more robust for changes in topology.

4. Operations and Characteristics of The SDD Protocol

The link layer must be able to automatically perform neighbor discovery to find the nodes that are in-range, build up connections to these nodes and form a link topology. The link layer also needs to control the time of transmission, specify the channel to be used for the transmission and maintain the neighbor list. To achieve this, we proposed in [3] the SDD protocol, a self-organizing link layer protocol on top of a TH impulse radio physical layer. In this section, we discuss performance related aspects of the SDD protocol. For more details on the operation of the protocol we refer to [3].

4.1. ASSUMPTIONS

We assume all nodes have omni-directional antennas and errorless channels. Each node in the network is supposed to know its own codes. Further, the UWB devices consume very little power when they are monitoring a code [6].

4.2. OVERALL DESCRIPTION OF THE SDD PROTOCOL

We consider a scenario where a number of nodes within radio range of each other are expected to form and maintain a single-hop ad hoc network. All nodes join or leave the network as a

result of being powered on or off, or by entering or leaving the radio range of the network nodes. The initial situation of each node is that it has no prior knowledge about its surroundings.

The SDD interfaces with the physical layer and the network layer. A collision indication, the codes and the protocol data units are communicated via the physical layer interface. The neighbor list including link metrics can be used by network layer routing protocols.

The protocol consists of two sub-processes: *Device Discovery* (DD) and *Data Transmission* (DT). When a node is powered on, it is able to establish direct links by using DD processes. When a node loses most of its links to its neighbors, this is an indication that the topology of the network has significantly changed; the node then needs to execute a DD again. The DD can be repeated with high or low frequency. The discovery happens with high frequency when a device is powered on or when there are strong indications that the topology has changed. The data transmission process executes whenever there is a request from the network higher layers of a node to send data to its neighbor(s). The DD and DT processes share the radio channel resources.

4.3. NEIGHBOR LIST

Every node keeps a list of the transmission codes of its neighbors. The number of neighbors for a node is the number of its related links, which is the degree of the node, denoted by d_i [5]. The maximum degree should always be equal or smaller than the maximum size of the neighbor list l_{\max} . For a given network, the maximum number of links that the MAC of a node can support is limited by l_{\max} .

The neighbor list is updated whenever a node discovers a new node or detects the disappearance of a neighbor; it records the codes and MAC addresses, etc., of its neighbors. In addition, in order to avoid an explicit removal procedure, a code in the neighbor list expires after it has not been used for a period longer than a timeout T_{\max} .

4.4. DEVICE DISCOVERY PROCESS

The DD process is triggered when a node is powered on and it has no knowledge about its surroundings. It ends when the node is powered off. This process is designed to reduce the time spent on synchronization. It discovers multiple nodes per synchronization. This is unlike Bluetooth (see Section 1) and it clearly speeds up the network formation. For example, Bluetooth needs up to 10.24s to finish one discovery cycle; on average, it spends 5–6 s finding only one device (see Bluetooth specifications). We will show that (see Section 5.3) our protocol uses around 50 ms to find 10 nodes and 70 ms to find 50 nodes.

We use an example with two nodes i and j to illustrate how the process works. To simplify the explanation, the propagation delay is not shown in the following figures. As shown in Figure 2, when node i is switched on, node i first stays in *Idle* state for a random period with uniform distribution. The process starts when node i broadcasts an Inquiry (IS) packet on the common code C . The acquisition header of the IS packet allows node j , which is in range and listening, to synchronize with node i . After sending the IS packet, node i starts a response scan operation for receiving responses from the inquired nodes. The number of slots during the response scan is called the *response scan window size* $w_{i,\text{scan}}$. During this operation, node i will periodically send Low Rate Synchronization (LRS) packets on code $C_{i,T}$ of node i to keep all inquired nodes synchronized.

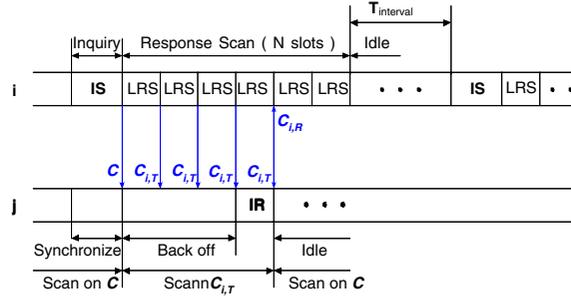


Figure 2. Device Discovery process.

If the link $i-j$ already exists, j does not have to reply and returns to *Idle* state monitoring code C . Otherwise, after a random number of time slots, denoted by $T_{j,backoff}$, node j responds with an Inquiry Response (IR) packet on the code $C_{i,R}$ of node i . $T_{j,backoff}$ is determined by node i 's scanning window size $w_{i,scan}$. It is uniformly distributed in the range $[0, w_{i,scan}]$. Afterwards, node j returns to *Idle* state. If node j is successful in sending the IR packet containing node j 's code $C_{j,T}$, node i and j have discovered each other and are ready for data transmission.

Response scan window size adaptation. A node i adjusts its response scan window size $w_{i,scan}$ based on the reception statistics during the current response scan process. The adjusted $w_{i,scan}$ will be used for the next DD process. By using adapted response scan window, the protocol can be suitable for some extreme initial configuration of a system. In any case, a system is able to self-configurable to optimal parameters all by itself.

The result of the response scan in every time slot as observed by node i can be *null*, *success* or *collision*; *null* implies that no IR packet was received nor a collision occurred, *success* means that an IR packet was successfully received and *collision* means that a collision was detected. Node i counts the number of slots with each result during $w_{i,scan}$ time slots. At the end of the response scan operation, if the number of *null* results is much larger than the other two, it means that the response scan window size $w_{i,scan}$ is considered too large for the number of in-range nodes and can be reduced without significantly increasing the probability of collisions. If the number of *success* results is dominating, it probably implies that $w_{i,scan}$ is a suitable number. Finally, if the number of *collision* results is much larger than the others, it indicates that $w_{i,scan}$ is too small and should be increased for the next device discovery process.

A proposed *adaptation algorithm* used in the simulation is described as follows:

$$\begin{aligned}
 &\text{if } (N_{\text{null}} > 2/3 \times w_{i,scan}) \\
 &\quad w_{i,scan} = \max(w_{i,scan}/\beta, W_{\min}); \\
 &\text{else if } (2 \times N_{\text{collision}} > N_{\text{success}} \\
 &\quad \text{or } N_{\text{collision}} > 1/5 \times w_{i,scan}) \\
 &\quad w_{i,scan} = \min(w_{i,scan} \times \beta, W_{\max});
 \end{aligned}$$

W_{\min} and W_{\max} are respectively the lower and upper bound of $w_{i,scan}$. The parameters of the algorithm are initially chosen and perform relatively better than a few alternatives. The choice of parameters will be determined experimentally through simulations.

Discovery interval adaptation. The repetition of the DD processes is further controlled by a *discovery interval* timer with adaptive time values. The *discovery interval* can be a variable with a short mean, denoted by $T_{\text{int_short}}$, or one with a long mean, denoted by $T_{\text{int_long}}$. The

short mean is used when a device is powered on to form a network or join a network while the neighbor list is empty, or when the destination is not in the neighbor list, or when a node has reason to believe it has lost its connectivity with its surroundings, e.g., due to power being turned off. The mean of the interval distribution changes from the short one to the long one in two cases. First, when the neighbor list reaches the maximum size l_{\max} ; second, when no new neighbor has been found in u successive short interval DD processes while u was set to 3 in [3].

Inter-DD scheduling. When one DD process is completed, one node adjusts its *discovery interval* to schedule next discovery. During this interval, the node can receive the IS packets sent by other neighboring nodes. Once the node receives an IS packet, it has to wait for a random back-off time before it replies with IR. Therefore, it happens sometimes that the back-off time doesn't expire when discovery interval expires. One way to solve this problem is that the next DD process only starts after the back-off time expires.

4.5. DATA TRANSMISSION PROCESS

The DT process is initiated by an RTS-CTS handshake using code C . Unicast and multicast are both supported.

Figure 3 illustrates the unicast data transmission process for two nodes. Let us assume that node i needs to send data to node j . As shown in Figure 3, if node i is *Idle*, it first broadcasts an RTS packet on code C . Without waiting for the response, node i continues to send an LRS packet on code $C_{i,T}$ to keep node j synchronized. Right after node j receives the RTS packet, it replies to node i by sending a CTS packet on code C . In the meantime, node j begins to monitor code $C_{i,T}$. Subsequently, the data packet is transmitted on code $C_{i,T}$ from node i to node j . After the data packet is transmitted, node j replies to node i with an ACK packet on code $C_{j,T}$. Finally both nodes i and j return to *Idle* state.

Data retransmission. When node i does not receive a CTS packet from node j , node i will periodically attempt to resend an RTS packet in T_s interval for at most m times. This is illustrated in Figure 3. If there is still no response after m number of RTS packets, it means that either node j has left the radio range of node i , node j has been powered off, or the connection between node i and j has been disturbed for a long time.

Multicast. MAC layer multicasting refers here to the process of sending a packet to some or all of the neighbors of a node. In the multicast operation of the SDD protocol, the codes of all the destination nodes are included in one multicast RTS packet. By monitoring the code

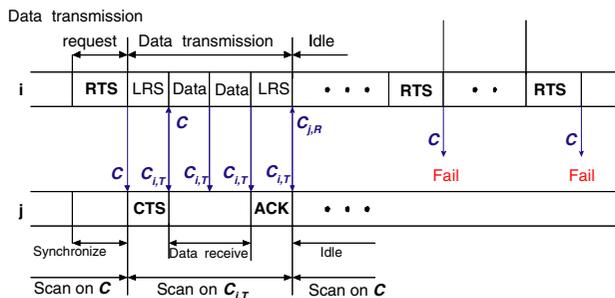


Figure 3. Data Transmission process.

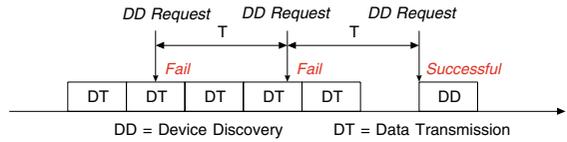


Figure 4. The coordination of DD and DT.

of the source node, the destination nodes can simultaneously receive the multicast packets. In the current version of the SDD protocol a multicast is not acknowledged by the destinations.

Coordination of DD and DT. Although, the DD and DT are separate processes, they require the use of the common codes. When one process is ongoing, the other should not interrupt and has to wait until the resource is released.

We explain the solution for the first situation that a DT process tries to operate while a DD process is ongoing. At the end of the DD process, the node needs to check if there is any buffered data packet to be transmitted. Thus, the node can deduce if it has to operate DT right after.

As illustrated in Figure 4, the second situation is that a discovery interval expires while a DT process is ongoing. The solution is to execute DD process after all the continuous DT processes have been finished. Thus, the node will hold on the DD process and periodically check until the resource is released.

5. Performance Evaluation

5.1. SYSTEM PARAMETERS

Let us discuss the system parameters we expect to have influence on the performance of the SDD protocol.

Neighbor list size l_{\max} . The neighbor list size l_{\max} is the maximum number of discovered neighbors that a node can store which is limited by system capability. When l_{\max} is large, a node is allowed to connect more nodes. When each node in the network knows large number of neighbors, more routes possibility is provided to network layer and the network is more robust. However, the complexity of route algorithm is increased. Therefore, the choice the maximum size of neighbor list is a trade-off between robustness and complexity.

Response scan window parameters. When a node receives an IS, it should first back-off for a random duration before responding. This back-off time is uniformly distributed over the response scan window $w_{i,\text{scan}}$. The larger the scan window size, the lower the probability of response packet collisions but on the other hand the longer the response scan duration. Two parameters are related to $w_{i,\text{scan}}$, of which one is the initial value of the scan window size W after a node is powered on and the other is the adaptation factor β . β is to scale the changes of $w_{i,\text{scan}}$ each time..

DD interval $T_{\text{int_short}}$ and $T_{\text{int_long}}$. The discovery interval determines the frequency with which the DD process is executed. The shorter the interval is, the more frequent the discovery process is executed, but potentially the more bandwidth is consumed.

5.2. PERFORMANCE PARAMETERS

5.2.1. Discovery time

The *discovery time* $T_{i,\text{disc}}$ is an important measure of the self-organization process for a node. It is defined as the duration of a set of successive DD processes with short average interval. It is counted from the moment that a node using $T_{\text{int_short}}$ interval to the moment it changes to $T_{\text{int_long}}$ interval mean (see Figure 5).

The period of one DD attempt is denoted as $T_{i,\text{dd}}$:

$$T_{i,\text{dd}} = v_{\text{IS}} + w_{i,\text{scan}} \times \sigma \quad (1)$$

where v_{IS} is IS packet transmission time and σ is the time slot used in response scan window. The average discovery time \bar{T}_{disc} is as follows:

$$\bar{T}_{\text{disc}} = \frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} \left(\sum_{k=1}^r T_{i,\text{dd}}(k) + \sum_{k=1}^{r-1} T_{\text{int_short}} \right) \quad (2)$$

There are k ($1 \leq k \leq r$) times DD attempts within the discovery time for a single node. r is obtained from the simulation. The shorter the average discovery time is, the more efficient the discovery protocol is.

When the discovery time for each node is considered, node join time $T_{i,\text{join}}^j$ and node depart time $T_{i,\text{depart}}^j$ are the interesting parameters. For any individual node j joining a node i , e.g., after power on, the time needed to become the neighbor of node i is also of interest, since it determines how fast node j can start communicating with node i . The lower and upper bound of node join time $T_{i,\text{join}}^j$ can be shown as follows:

$$T_{i,\text{join}}^j \geq T_{\text{ini}} + w_{\text{ini}} \times \sigma \quad (3)$$

where T_{ini} is the initial period after a node is powered on. Although we did not investigate this measure, it is strongly related to the \bar{T}_{disc} .

The node depart time is related to the random retransmission period. After maximum m times of RTS packets retransmission, if there is still no response from node j , node i decides that data transmission has failed. This time can be used for node i to decide that the neighbor j has left.

$$T_{i,\text{depart}}^j = \sum_{k=1}^m T_s + m(v_{\text{RTS}} + v_{\text{LRS}}) \quad (4)$$

where v_{RTS} and v_{LRS} are packet transmission time of RTS and LRS packets.

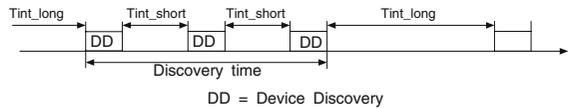


Figure 5. SDD discovery time.

5.2.2. Discovery Ratio

Discovery ratio defined in SDD protocol is calculated as the average percentage of discovered neighbors after one *discovery time*.

$$\bar{\rho} = \frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} \frac{d_i}{N_{\text{total}} - 1} \quad (5)$$

where d_i is the degree of any node i .

5.2.3. Effect on Throughput

We measure the achievable throughput when DD process is performed as well as when DD process is not performed. The comparison of the two measurement results can show the effect of DD process on throughput.

Throughput S is defined as the average successful transmission rate, or

$$S = \frac{\text{Successfully received bits}}{\text{Simulation time elapsed}} \quad (6)$$

5.3. SIMULATION RESULTS

We did an initial performance evaluation of the SDD protocol using GloMoSim [8], which is a discrete event simulator. A SDD module has been developed and embedded in GloMoSim.

Based on an interactive game scenario described in [2], we specify the following parameter settings for the experiments. We consider a number of nodes forming a single-hop network within an area of $10\text{ m} \times 10\text{ m}$, i.e., corresponding to an in-room environment. The number of nodes is less than 50. The nodes are uniformly distributed over the simulation area. Node mobility is neglected in this scenario, i.e., after the initial placement, each node will stay stationary from the beginning to the end of the simulation. We assume an error-free channel, hence all packet losses are due to collisions or buffer overflow.

The system parameters used to evaluate the SDD protocol are listed in Table 1. We set the achievable data rate to 110 Mbps corresponding to the IEEE 802.15.3a. This capacity is kept constant since in the experiments we limit the number of simultaneous communications to 10. The signal acquisition time is set to 0.1 ms, which can be further improved [11]. In order to avoid frequent synchronizations, we use a large packet length of 2000 bytes. The traffic model we use is a constant bit rate (CBR) model.

The choice of protocol parameters is mostly based on preliminary experiments.

We designed four experiments to evaluate the performance of the SDD protocol in order to study the effect of the system parameters on *average discovery time* \bar{T}_{disc} , *discovery ratio* $\bar{\rho}$ and the effect of DD process on *throughput*, as we discussed in Section 2. In the first three experiments, we assume a number of nodes powered on at the beginning of the simulation and perform a DD process without being interrupted by data transmission. In the last experiment, the data transmission is involved and measured.

Table 1. Parameter specification

Parameter	Value
Achievable data rate	110 Mbps
Traffic model	CBR
Packet size	2000 bytes
Buffer size	10000 packets
Channel acquisition time	0.1 ms
IS packet size	36 bytes
IR packet size	20 bytes
RTS packet size	36 bytes
CTS packet size	36 bytes
Initial time T_{ini}	[1 ms, 20 ms]
Slot duration for device discovery σ	20 μs
Discovery short interval $T_{\text{int_short}}$	10 ms
Discovery long interval $T_{\text{int_long}}$	1 s
Retransmission interval mean $E[T_s]$	0.2 ms
Retransmission times m	3
Response scan window initial value W	15 or 32 slots
Neighbor list size l_{max}	10 – 100 entries
Neighbor list time out T_{lmax}	100 s

5.3.1. Discovery Time

We first investigate the influence of the response scan window size on *discovery time*. We assume l_{max} is larger than the number of nodes and hence does not impact the discovery time. In Figure 6, we observed a certain degree of linearity in terms of the number of nodes. Moreover, the maximum discovery time for a 50-node setting is less than 75 ms which is much lower than the magnitude of the discovery time in Bluetooth.

We also observe that as we use different initial scan window size W and the adaptation factor β (see Figure 6), \bar{T}_{disc} is slightly different within a certain range. The difference is caused by the different adaptation parameters W and β . The difference is not so dramatic because in the initial setting, $T_{\text{int_short}}$ is one or two orders of magnitude larger than the response scan window period which is from 0.2 ms to 0.7 ms.

We now investigate the influence of the neighbor list size l_{max} on average discovery time \bar{T}_{disc} for different numbers of in-range nodes. In this experiment, the initial scan window size W is chosen as 31, and the adaptation factor β is 2.

As shown in Figure 7, we can see that when the neighbor list size l_{max} is smaller than the number of nodes, the discovery time increases as l_{max} increases. The reason is that the DD processes with $T_{\text{int_short}}$ interval terminate only according to one condition that the neighbor list is full. More time is needed by each node to fill in its neighbor list when l_{max} increases. In this case, the degree d_i of each node is equal to l_{max} .

If l_{max} is equal to or larger than the number of adjacent nodes, when the network is fully connected, the discovery time reaches a steady maximum value. This is because the DD processes with $T_{\text{int_short}}$ interval terminate due to the fact that no new neighbor has been found in the last three contiguous discovery processes.

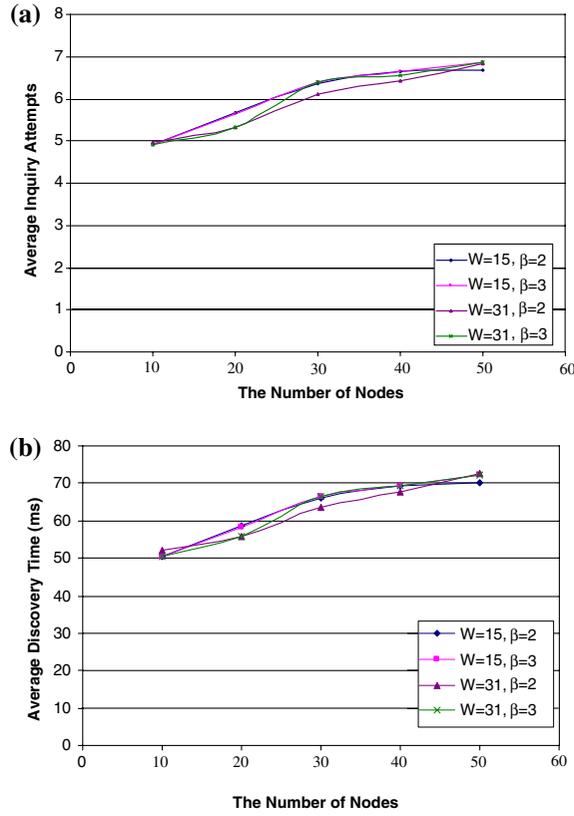


Figure 6. (a) The average number of DD processes in one discovery. (b) The average discovery time.

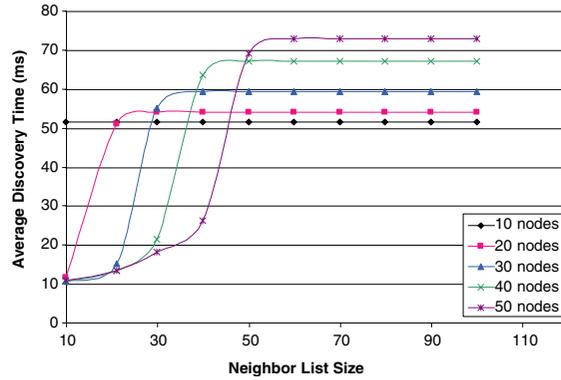


Figure 7. The average discovery time for different neighbor list sizes.

5.3.2. Discovery Ratio

With the same parameter settings as in the initial experiments, the average discovery ratio was measured. We set the neighbor list size l_{\max} much larger than the total number of node.

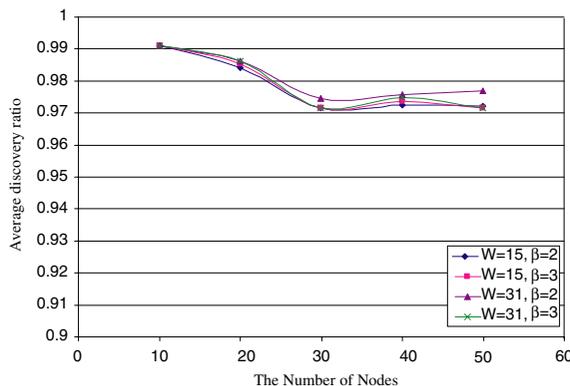


Figure 8. Average discovery ratio.

From Figure 8, we observe that the average discovery ratio $\bar{\rho}$ is always higher than 0.97 for different number of nodes. It shows that most nodes can find most of their neighbors in one discovery time. Thus, the degree of any node is very close to the number of in-range nodes. Therefore, the node similarity can be satisfied in the network topology formed by SDD protocol.

In Figure 8, when the number of nodes increases, $\bar{\rho}$ slightly decreases. This is because when more nodes come up in the network, the probability that more DD processes overlap with each other increases, as well as the probability of collisions of packets.

5.3.3. Throughput Behavior

As explained in Section 3, the device discovery and data transmission processes make use of the same radio resources, it is necessary to evaluate the influence of the DD processes on the system throughput.

The DD processes will be executed in three situations, i.e., the initial DD process after power is on, the periodical execution with the interval of $T_{\text{int_short}}$ or $T_{\text{int_long}}$. The execution due to retransmission failure of a data packet.

We only consider 4 nodes in this experiment and thus the effect of MAI and routing protocol can be neglected. The CBR traffic is sent from node 0 to node 1 and node 2 to node 3. The packet size and buffer size was listed in Table 1. In both the situations of without and with DD processes (see Table 2), the initial DD process has to be executed.

When the packet generation rate is smaller than 4000 pkts/s and the network is lightly loaded, the throughput increases with the packet generation rate (see Figure 9). For higher packet arrival rates, the network is saturated and the throughput stays above 54 Mbit/s. When the network is either lightly loaded or saturated, we observe that the device discovery has very little influence on the system throughput. Table 2 shows the difference in throughput without and with the DD process. It benefits from the high data rate and capability of multiple transmissions offered by TH impulse radio. The fast transmission of control packets on the common code reduces the probability of collision occurrence between the DD and the DT processes. The multiple transmissions on different codes enable the two simultaneous processes to be successfully executed.

Table 2. Throughput

pkts/sec*1000	Without DD (Mbit/s)	With DD (Mbit/s)
1	16	15.99
2	32	31.99
3	48.048	48.017
4	54.681	54.492
5	54.681	54.492
6	54.681	54.492
7	54.681	54.492
8	54.681	54.492
9	54.681	54.492

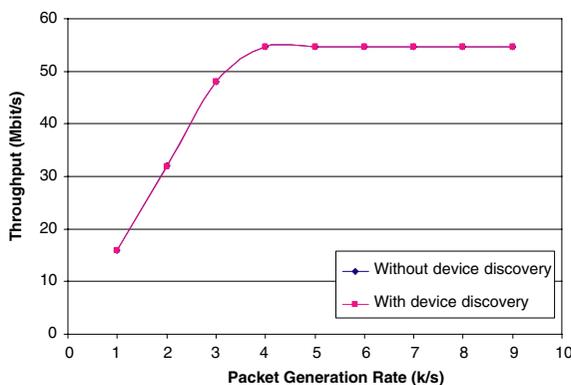


Figure 9. Throughput behavior with/without device discovery.

6. Conclusion

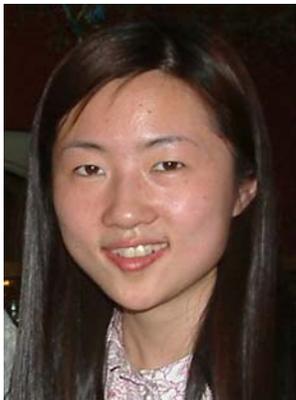
In this paper, we have defined performance criteria for designing a discovery protocol based on self-organization principles. The key characteristics and optimizations of a self-organizing link layer protocol (SDD) for impulse radio UWB were described. The SDD protocol was implemented in a GloMoSim simulation environment. Results show that the SDD protocol has good performance in the operations of neighbor discovery and medium access. In the current experiment conditions and assumptions, the average discovery time is much lower than the discovery time in Bluetooth. The node similarity can be satisfied in the single-hop network topology. The device discovery process has very little influence on the system throughput.

We are presently working on the multi-hop case and on performance analysis using different traffic models.

References

1. W. Lu, A. Lo and I. Niemegeers, "On the Dynamics and Self-organization of Personal Networks", *MAGNET Workshop in Shanghai*, November, 2004.

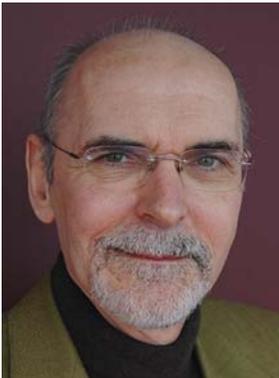
2. A.N. Zadeh and B. Jabbari, "Self-organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)", *Commun. Mag. IEEE*, Vol. 40, No. 6, pp. 149–157, June 2002.
3. N. Shi and I. Niemegeers, "A Self-organizing Link Layer Protocol in UWB Ad Hoc Networks", *Proceedings of Personal Wireless Communications: IFIP TC6 9th International Conference*, PWC 2004, Delft, September 2004, pp. 248–261.
4. <http://pcl.cs.ucla.edu/projects/glomosim/>.
5. A.H. Dekker and B.D. Colbert, "Network Robustness and Graph Topology", *Proceedings of the 27th conference on Australasian Computer Science*, Vol. 26, 2004, pp. 359–368.
6. J. R. Foerster and Q. Li, "UWB Channel Modeling Contribution from Intel", Intel Corporation, Hillsboro, OR, USA, Tech. Report P802.15 02/279 SG3a, June 2002, IEEE P802.15 SG3a contribution.
7. M.G.M. Hussain, "Principles of Space-time Array Processing for Ultrawide-band Impulse Radar and Radio Communications", *IEEE Trans Vehicular Technol*, Vol. 51, No. 3, May 2002, pp. 393–403.
8. S.S. Kolenchery, J. K. Townsend, and J.A. Freebersyser, "A Novel Impulse Radio Network for Tactical Military Wireless Communications", *Proceedings of IEEE Milcom '98*, Boston, October 1998.
9. M.Z. Win and R. A. Scholtz, "Ultra-Wide Bandwidth Time-Hopping Spread-Spectrum Impulse Radio for Wireless Multiple-Access Communications", *IEEE Trans. Commun*, Vol. 48, No. 4, April 2000, pp. 679–689.
10. L. Zhao and A.M. Haimovich, "The Capacity of an UWB Multiple Access communication System", *Communications, 2002. ICC 2002. IEEE*, Vol. 3, 28 April–2 May 2002, pp. 1964–1968.
11. R. Djapic, G. Leus and A.-J. van der Veen, "Blind Synchronization in Asynchronous Multiuser UWB Networks Based on the Transmit-reference Scheme", *The Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove CA, November 7–10, 2004.
12. I. Oppermann, "The Role of UWB in 4G", *Special Issue on Adaptive Global Net: Vision Towards a Modern Communication Infrastructure*. *Wireless Personal Commun.* April 2004, Vol. 29, No. 1–2, pp. 121–133(13).
13. J. Ding, L. Zhao, S. R. Medidi, and K. M. Sivalingam, "MAC Protocols for Ultra-Wide-Band (UWB) Wireless Networks: Impact of Channel Acquisition Time", *ITCOM*, 2002.
14. J. J. Garcia-Luna-Aceves, and J. Raju, "Distributed assignment of codes for multihop packet-radio networks", *MILCOM 97 Proceedings*, Vol. 1, November, 1997, pp. 450–454.



Nan Shi received Bachelor of Engineering from Beijing Institute of Technology in China in 1998. In 2002, she obtained Master of Science degree in Information Communication Technology (ICT) at Faculty of Electrical Engineering in Delft University of Technology. Since September 2002, she has been working as a doctoral researcher in Wireless and Mobile Communications (WMC) Group, Delft University of Technology. Her research mainly covers the area of architecture and protocol design for UWB-based ad hoc networks. Currently, her research involves the issues on device discovery, medium access control and multi-hop forwarding in impulse radio based networks.



Liang Xia received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China in 2001, and the M.Sc. degree from Delft University of Technology, the Netherlands in 2004, both in Electrical Engineering. Since September 2004, he has been working toward the Ph.D. degree in Electrical Engineering at Delft University of Technology, the Netherlands. His current research interests include ultra wideband positioning and communications, medium access and routing in mobile and hoc networks and wireless sensor networks.



Ignas G.M.M. Niemegeers got a degree in electrical engineering from the University of Gent, Belgium, in 1970. In 1972 he received an M.Sc.E. degree in computer engineering and in computer engineering and in 1978 a Ph.D degree from Purdue University in West Lafayette, Indiana, U.S.A. From 1978 to 1981 he was a designer of packet switching networks at Bell Telephone Mfg. Cy. Antwerp, Belgium. From 1981 TO 2002 he was a professor at the Computer Science and the Electrical Engineering Faculties of the University of Twente, Enschede, the Netherlands. From 1995 to 2001 he was Scientific Director of the Centre for Telematics and Information Technology (CTIT) of the University of Twente, a multi-disciplinary research institute on ICT and applications. Since May 2002 he holds the chair Wireless and Mobile Communications at Delft University of Technology, where he is heading the Centre for Wireless and Personal Communication (CWPC) and the Telecommunications Department. He is an active member of the Wireless World Research Forum (WWRF) and EFIP TC-6 Working Group on Personal Wireless Communication. He was involved in many European research projects, in particular ACTS TOBASCO, ACTS PRISMA, ACTS HARMONICS, RACE MONET, RACE INSIGNIA and RACE MAGIC. His present research interests are Beyond 3G wireless infrastructures. UWB networks, ad-hoc networks, Personal Area Networks and Personal Networks, Mobile IP and ubiquitous computing and communication.